Europäisches Patentamt | European Patent Office | Office européen des brevets

EPO - DG 1

EP 00/2274

REC'D 21 JUN 2000 07. 06. 2000

WIPO PCT

65

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

99200930.8

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE 23/05/00

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99200930.8

Anmeldetag:
Date of filing:
Date de dépôt: 26/03/99

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Digital rights management for solid state audio players

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Digital rights management for solid state audio players.

Enabling memory modules to play in any compliant player using peer-to-peer authentication and exchange of a dynamic memory ID.

The fast-track goal of the Secure Digital Music Initiative (SDMI) is to specify the security technology for portable players. It is required that this technology protects the interests of the content providers, be it the major record labels or small garage bands, yet addresses consumer interests such as convenience, sound quality, and privacy as well.

For consumer convenience, it must be possible to store protected audio on one player, and have it play on another player (by transferring the detachable memory module), without the need for an on-line transaction, or for the two players to be physically connected. Of course, it should not be possible to make perfect bit copies of the modules such that the duplicates both play simultaneously in different players. The straightforward solution to this requirement is to encrypt the audio using a property of the module which is unique for each module and which cannot be changed. For example, one could encrypt the audio using a key which is derived from a "module ID" (which in a passive memory module is a fixed and public number), and a secret key which is globally shared by all players.

A more flexible scheme is obtained by the memory module architecture proposed earlier (see patent proposal PHN 17.357). The core of this proposal is that each sector on the memory module has associated a field (dubbed the "Secure Solid Sate Sector Tag" or S4T) which is to hold a random number. This random number is to be renewed on each write operation to that sector by some dedicated (preferably on-chip) logic, and has read-only access to devices employing the module. It has been explained that this means can be used to prevent replay attacks, by encrypting the audio using a key which is derived from the random numbers associated with the sectors in which it is stored, and the globally shared secret as mentioned above.

From a security point of view, the use of a globally shared secret is not desirable as it makes the system vulnerable to isolated attacks. Namely, once the globally shared secret is compromised, the protected content on all players can be easily decrypted (by other unauthorised means). Therefore, since SDMI call for proposals includes a requirement that "the system should be designed such that isolated attacks do not result in systemic failures," such a globally shared secret cannot be used. The purpose of the current proposal therefore is to find a way which allows exchanging of memory modules between players, without using global secret, and without inconveniencing the consumer too much. It is meant as a mechanism for players which have infrequent needs to exchange audio content. Players which need to be able to exchange content on a regular basis can be registered to the "Licensed SDMI-Compliant Module" (LCM) as defined by the SDMI Call for Proposals. This should provide a means to give all players of the group access to the content, without the need for an authentication mechanism such as described below.

The idea is to introduce a "dynamic memory ID," which stored on a blank memory module by the first player in which the module is used. This dynamic memory ID is protected using the S4T fields and public key cryptography. When using the module in a second player for the first time, that player should store its signature on the module and request access to the dynamic memory ID. Access is granted by the first player on re-insertion of the module. Subsequently, the second player can play the audio stored on the module as well as store audio on the module which is playable by both the second and the first player. The only user inconvenience involved is that the module should be re-inserted once in the first player for authentication, before it can be used in the second player. If the module is to be used in a third player, this procedure is to be repeated using either the first or the second player as an authenticating device.

To implement this scheme it is required to have the following items available in each player:

- a public player ID;
- a public certificate vouching for the player's public key; and
- a secret private key.

Clearly, the only secret present in a player is its private key. This means that isolated attacks against a single player which compromise that player's private key will not lead to a system

wide failure. The system wide (globally shared) secret is the private key of the certifying entity, which easily can be kept in a safe place.

Figures 1–4 show the operation of the proposed system in more detail. Four steps can be distinguished:

1. Player A creates and signs the dynamic memory ID on an empty module. Next, it stores some audio content and associated usage rights on the module. Note that this is not an actual step in the process, but the starting point for the authentication and exchange.
2. The module is inserted in player B, which stores its public key certificate on the module.
3. The module is re-inserted in player A, which makes the dynamic memory ID available to player B using that player's public key.
4. The module is re-inserted in player B, which verifies the dynamic memory ID, and if correct, removes the references to player A from its copy of the dynamic memory ID, and signs it.

The symbols used in these figures have the following meaning:

- ID_A the device ID of player A;
- $K_{Priv,A}$ the private key of player A;
- $K_{Pub,A}$ the public key of player A;
- $Cert(K_{Pub,A})$ player A's public key certificate;
- S the (secret) dynamic memory ID;
- Enc[K;...] encryption with public/private key K;
- E[K; ...] encryption with key K using a block cipher;
- H[...] a cryptographic hash of the arguments;
- T_I are the random S4T values; and
- p, q, x, y are random numbers.

Figure 1 shows the contents of the memory module after the first step. A dynamic memory ID S has been created and stored on the module, after encryption with the public key of player A. Therefore, S is available to player A only. To ensure that this dynamic memory ID cannot be reused on the current or a different memory module (replay attack), a hash of S and T_1 is signed using the player's private key. Next the encrypted audio content and the associated rights and usage information is stored. The key K which has been used to

encrypt the audio itself is encrypted with a hash of the S4T values, and the dynamic memory ID S.

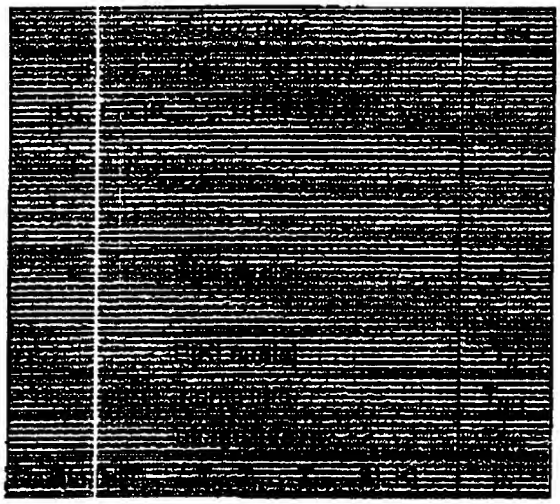

Figure 1: Contents of the memory module after the first step.

Figure 2 shows the contents of the memory module after the second step. The module has been inserted in player B, which has stored its public key certificate on the module.

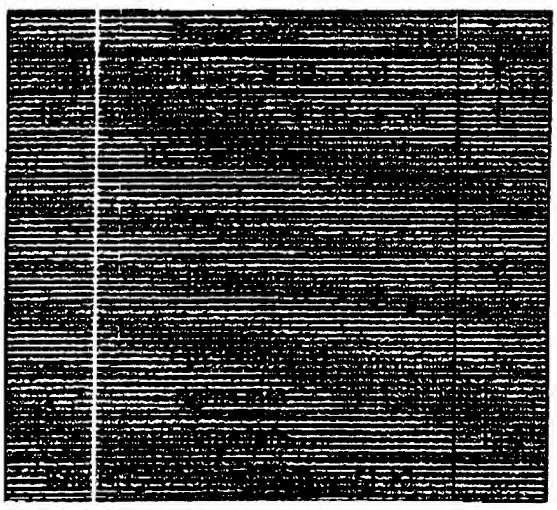

Figure 2: Contents of the memory module after the second step.

Figure 3 shows the contents of the memory module after the third step. Now the module has been re-inserted into player A, which has created a copy of the dynamic memory ID for player B's private use by encrypting S with B's public key. To ascertain that B knows that player A has provided S, A's ID and the tags associated with A's copy of S are included in the message to player B as well.

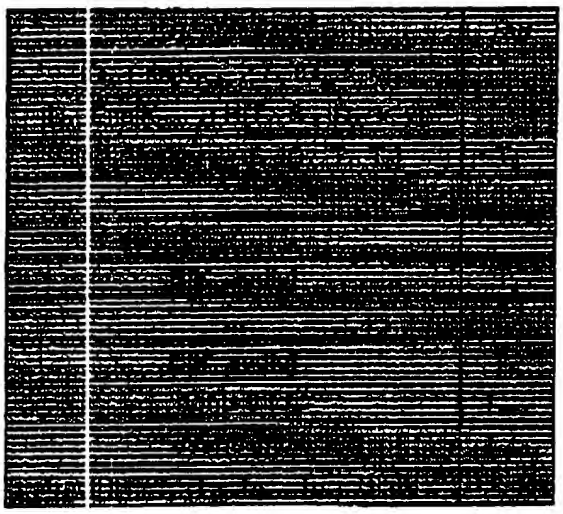

Figure 3: Contents of the memory module after the third step.

Figure 4 shows the contents of the memory module after the fourth step. The module has been re-inserted into player B, which is now is able to obtain S. In addition, player B can check that it is indeed a compliant player which has delivered S by testing A's ID and the tags against the memory contents. If the test is successful, player A is authenticated, and S is accepted. Subsequently, player B replaces the message from player A with its own copy of the dynamic memory ID (including signature), such that the format is identical to that of A's copy. Accordingly, it can not be traced any longer which player was used originally to store the audio content on the module. This helps to ensure privacy.

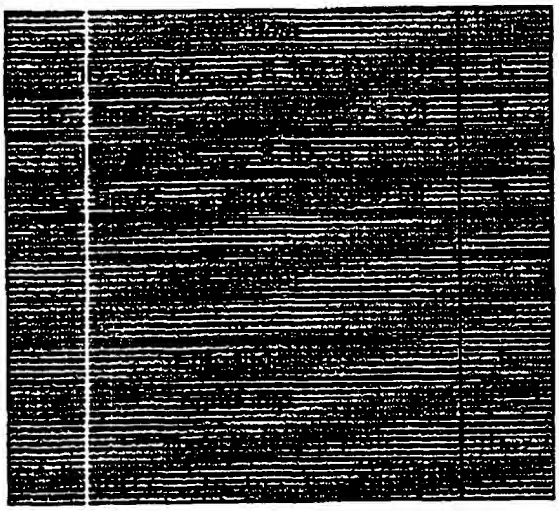

Figure 4: Contents of the memory module after the fourth step.

After these steps, both players have access to the audio content on the module through their private copy of the dynamic memory ID. As such, both will be able to play without additional handling by the user. However, bit copying the module while retaining play back capabilities is prevented by means of the previously described S4T mechanism. On play back, a player checks its ID against the list of authenticated players (A and B in the example). If it is present in the list, the dynamic memory ID is decrypted and checked against the signature which is required to present as well. Then, if all terms have been met, play back can start with decryption of the content key.

Addendum: exchange of modules within a group of players

In this case the idea is that all players which should form a group register their public key and corresponding certificate with the LCM. Therefore, if the LCM is employed to store some audio content on the memory module, it can directly make the dynamic memory ID accessible to all players within the group: it already has knowledge of all relevant public keys and player IDs. Figure 5 shows the resulting module contents. It is very similar to the state obtained using the peer-to-peer mechanism. The only difference is that the dynamic memory ID is now signed by the LCM rather than by the player. The advantage of creating a group may be clear: memory modules may be exchanged freely between members of the group, without the need for an authentication mechanism.

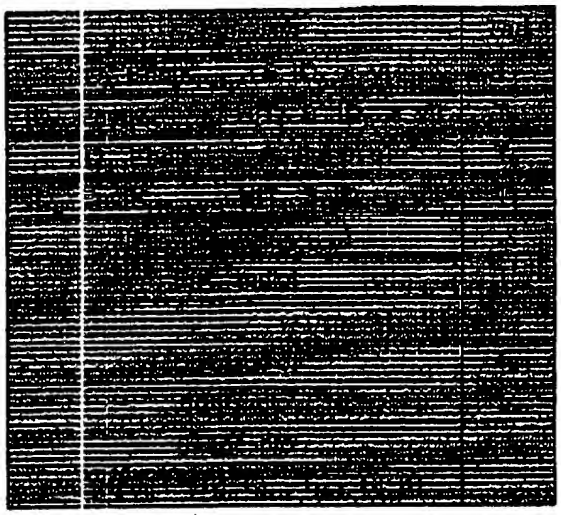

Figure 5: Contents of the memory module in case of a group

CLAIMS:

1. A method of copy protection substantially as described herein with reference to one or more of the accompanying drawings.

2. A method of exchanging memory modules substantially as described herein with reference to one or more of the accompanying drawings.

3. A device substantially as described herein with reference to one or more of the accompanying drawings.

4. A device wherein a method as claimed in Claim 1 or 2 is used for copy protecting the content stored in the device.

5. A device wherein a method as claimed in Claim 1 or 2 is used for exchanging memory modules.